Ingeniería y Ciencia
ISSN:1794-9165 | ISSN-e: 2256-4314
ing. cienc., vol. 15, no. 29, pp. 103–125, enero-junio. 2019.
http://www.eafit.edu.co/ingciencia

Enhanced RC5 Algorithm Using Parallel Computing for Communication Networks

David Fabián Cevallos Salas[1]

Received: 10-02-2019 | Accepted: 15-05-2019 | Online: 31-05-2019

MSC:14G50, 94A60

doi:10.17230/ingciencia.15.29.4

Abstract

One of the main disadvantages of symmetric key algorithms in comparison with asymmetric key ones is their lower security level due to their shorter key length. Indeed, increasing the number of bits that conforms the key used by a symmetric cryptography algorithm will increase its security level with a cost on its performance. Expanding a key in symmetric cryptography is not an easy task due to algorithms are designed to work with keys of a fixed maximum length. This paper presents an alternative design of the RC5 cryptography algorithm with an enhanced security level achieved through a key expansion mechanism with Parallel Computing processing. Although the design was created for the RC5 algorithm the main idea might be applied to other block cipher algorithms applying the same criteria. This methodology makes feasible to obtain a robust symmetric key algorithm implemented in software with an acceptable performance in comparison with other techniques such as implementations in hardware, reduction in the amount of data, improvements in the key exchange process, advanced high performance computing, and many others techniques.

Keywords: Cryptography; symmetric key; RC5; algorithm; performance.

[1] Universidad Tecnológica Equinoccial, Facultad de Ciencias de la Ingeniería e Industrias, davidf.cevallos@ute.edu.ec, http://orcid.org/0000-0002-3098-3090, Quito, Ecuador.

Algoritmo RC5 mejorado usando computación paralela para redes de comunicaciones

Resumen

Una de las principales desventajas de los algoritmos de clave simétrica en comparación con los asimétricos es su menor nivel de seguridad debido a su longitud de clave más corta. Ciertamente, aumentar el número de bits que conforman la clave utilizada por un algoritmo de criptografía simétrica aumentará su nivel de seguridad con un costo en su rendimiento. Extender una clave en criptografía simétrica no es una tarea fácil ya que los algoritmos están diseñados para trabajar con claves de una longitud máxima fija. Este artículo presenta un diseño alternativo del algoritmo de criptografía RC5 con un nivel de seguridad mejorado que se logra a través de un mecanismo de expansión de clave con procesamiento de Computación Paralela. Aunque el diseño fue creado para el algoritmo RC5 la idea principal podría ser aplicada a otros algoritmos de cifrado en bloque aplicando el mismo criterio. Esta metodología hace factible obtener un algoritmo de clave simétrica robusto implementado sobre software con un rendimiento aceptable en comparación con otras técnicas como implementaciones sobre hardware, reducción en la cantidad de datos, mejoras en el proceso de intercambio de clave, computación avanzada de alto rendimiento, entre otras.

Palabras clave: Criptografía; clave simétrica; RC5; algoritmo; rendimiento.

1 Introduction

Symmetric key algorithms make use of shorter keys in comparison with the keys used by asymmetric algorithms such as RSA (Rivest, Shamir and Adleman), ElGamal or Elliptic Curve Cryptography (ECC) [1]. For instance, AES (Advanced Encryption Standard) uses a key of maximum 256 bits whereas RSA works with a minimum recommended key length of 1024 bits. As is described in [2], having a larger key means more security due to the encrypted information will be more resistant to a brute force attack. However, computing a larger key also means more time, energy and resource consumption, which could lead to a lost in performance [3]. This fact makes asymmetric key algorithms have a stronger security level with a lower performance

in comparison with symmetric key algorithms, which demonstrated in [1].

On the other hand, symmetric keys have a shorter live time than asymmetric keys, which means that more maintenance and key management need to be applied for the first ones.

Although symmetric key algorithms are faster than asymmetric ones, even more if they are implemented in hardware, the fact that the sender and the target need to have the key before the communication be established is a major disadvantage. Asymmetric key algorithms use two keys in order to avoid this problem, as [4] describes, turning them in the preferred option for electronic communications with web services over public networks, and limiting to symmetric key algorithms to certain tasks such as the initial interchange of cryptography material in VPN communications [5].

Finding out the balance between the adequate security level and an acceptable performance is a difficult task [6]. The perfect scenario is to achieve the performance of symmetric key algorithms with the security level of asymmetric ones. Although it is not feasible, at least with the actual technology, the goal is to extend the conventional symmetric key lengths for making the algorithms to work with a key as large as possible.

The main symmetric key algorithms are based on a technique called block ciphering, which consists of dividing the data into small pieces (blocks) of certain length (which depends on the algorithm features) for being processed [7]. This technique is usually combined with iterative rounds which improve the performance of the algorithm. Different logical operations are performed at each iteration depending on the algorithm. Stronger algorithms use more complex operations.

Because of this operation mode the main constraint with symmetric keys is that the maximum key length of the algorithms is of a defined number of bits, limiting its security level. As [8] emphasizes, this feature is exacerbated by the fact that just few symmetric key algorithms allows a variable key length until its threshold whereas the vast majority just allows certain values inside its working range.

One of the symmetric key algorithms with that feature is the RC5 (Rivest Cipher 5) algorithm. This algorithm is used in several application environments such as image encryption [9] and ad hoc networks [10], and its simple structure makes it ideal for implementing modifications. More than this, the algorithm uses basic logical operations and has a simple structure making it easy of understanding [11].

As other symmetric key algorithms, RC5 can be implemented with basic hardware and software. It is possible to implement the algorithm with FPGA hardware as is made in [12].

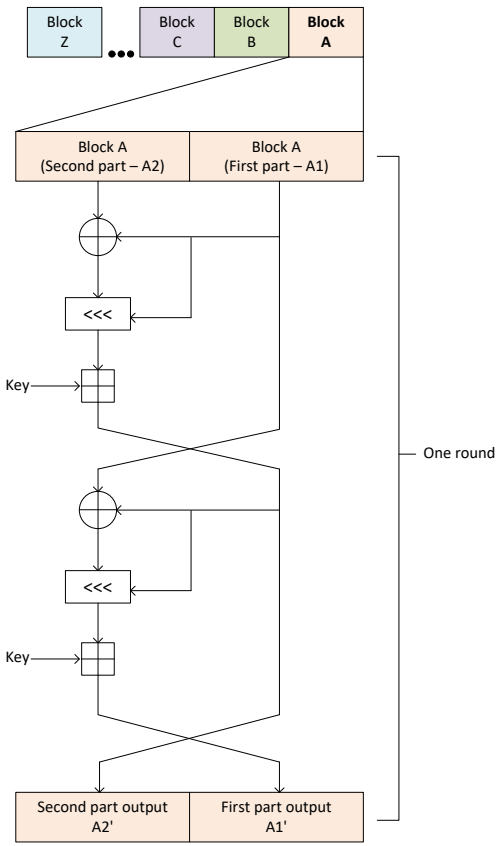


Figure 1: Basic RC5 encryption method

RC5 is derived from the RC4 algorithm and constitutes the basis for the actual RC6 algorithm. RC5 has a variable block size of 32, 64 and 128 bits, with a variable key length between 0 and 2040 bits. In [13] is demonstrated that for 12-round nominal RC5 version with a 64 bits block length, the algorithm is vulnerable to linear cryptanalysis.

Figure 1 explains the basic RC5 encryption method which will be used for enhancing the security level of this symmetric key algorithm without a reduction in its performance.

2 Enhanced RC5 algorithm using parallel computing for communication networks

In order to improve the performance of a symmetric key algorithm several techniques and methods can be applied; not only implemented in hardware, but also using software and even embedded systems with associations between hardware and software [14]. Although not all of these techniques allows to rise the security level, they can be applied together with other mechanisms to achieve a better security level with an acceptable performance.

One technique involves the creation of a new hardware platform oriented to enhance the speed through modern electronics systems [15]. This is the most common technique employed since symmetric key cryptography algorithms run faster in hardware instead of software, and several architectures have been created for this purpose. In [16] a platform made with modern FPGA is explained. The gate array permits to process several logic operations simultaneously improving the performance. Another example is exposed in [17] where a reconfigurable architecture implemented on Xilinx FPGA is presented. Another similar work is implemented in [18].

Another technique is to improve the key exchange mechanism instead of the algorithm [19]. This might improve the overall performance of the encryption process, although not the algorithm security level itself. In [13] is exposed this technique applied for asymmetric key algorithms used for vehicular and ad hoc networks. The main

problem is the fact that asymmetric key cryptography exchange process always will be better due to the use of a public key which is known before the encryption process starts, unlike the case of symmetric key algorithms.

Reducing the number of rows the algorithm uses to work is another performance technique. The main idea is to reduce the load the process running the algorithm has to work with. Although it is a valid option, this method reduces the security level instead of improving it.

Decreasing the amount of data to be encrypted as is described in [20] is also valid to increase performance, but it has a similar problem that the previous explained technique.

The use of High Performance Computing is the best option when there are available computational resources. In [15] is presented a mechanism to improve significantly the performance of symmetric key algorithms using a peer to peer computational grid middleware schema. However, it might not be possible to employ this technique in all the environments where cryptography is required.

In [21] and [22] is demonstrated that using Parallel Computing is possible to improve the performance even for asymmetric key cryptography. A similar scenario is presented in [23] in an environment with multi-core processors.

Other advanced techniques are nowadays being researched such as the use of Quantum Computing to process big data [24]. Quantum Computing will permit to make cryptography mathematical operations with significant improved performance [25]. However, the use of Quantum Computing looks still far away.

Therefore, modifying the algorithm structure to work with a better performance might be the best option when there are limited resources. As [26] explains, it is feasible to make a symmetric key algorithm to work with improved logical operations that yields the same results as the traditional XOR operations, but in a minor time and with a better performance. This technique also requires additional hardware components in order to support the new set of instructions. In order to avoid hardware implementations the modified algorithm

can be implemented in software, although with less performance.

For instance, as is explained in [27], RC5 structure can be modified and implemented in hardware for making the algorithm to run faster. In [28] is presented a similar approach for achieving this, but implemented in software and with an enhanced security level accomplished through a key expansion technique in a randomly manner.

Expanding the key length is the base option in order to rise the algorithm security level, but this will need to work together with one of the techniques previously described that allow to enhance the performance due to the fact that a larger key will make the algorithm to run slower.

The use of Parallel Computing is the best option for this scenario because it can help to mitigate this issue since it allows to carry out several processes at the same time. However, if a key expansion technique is used with Parallel Computing, the algorithm also needs to be modified to work dividing the data into small parts to be performed by each process simultaneously.

Current communication networks need to establish security mechanisms able to guarantee data confidentiality with good performance. Enhancing the security level of symmetric key algorithms without a performance degradation permits to use several applications in a safe way over wireless or public communications networks, as is described in [29].

3 Methodology

In order to improve the RC5 algorithm security level is necessary to make use of a larger key length. The performance is not reduced due to the use of Parallel Computing.

3.1 Modifying the algorithm

The main idea is to create a second process which simultaneously will execute with parallelism the RC5 traditional algorithm for the

second block, while the first one is also executed; as is reflected in Figure 2. The combination of the blocks being processed is called an inter-block.

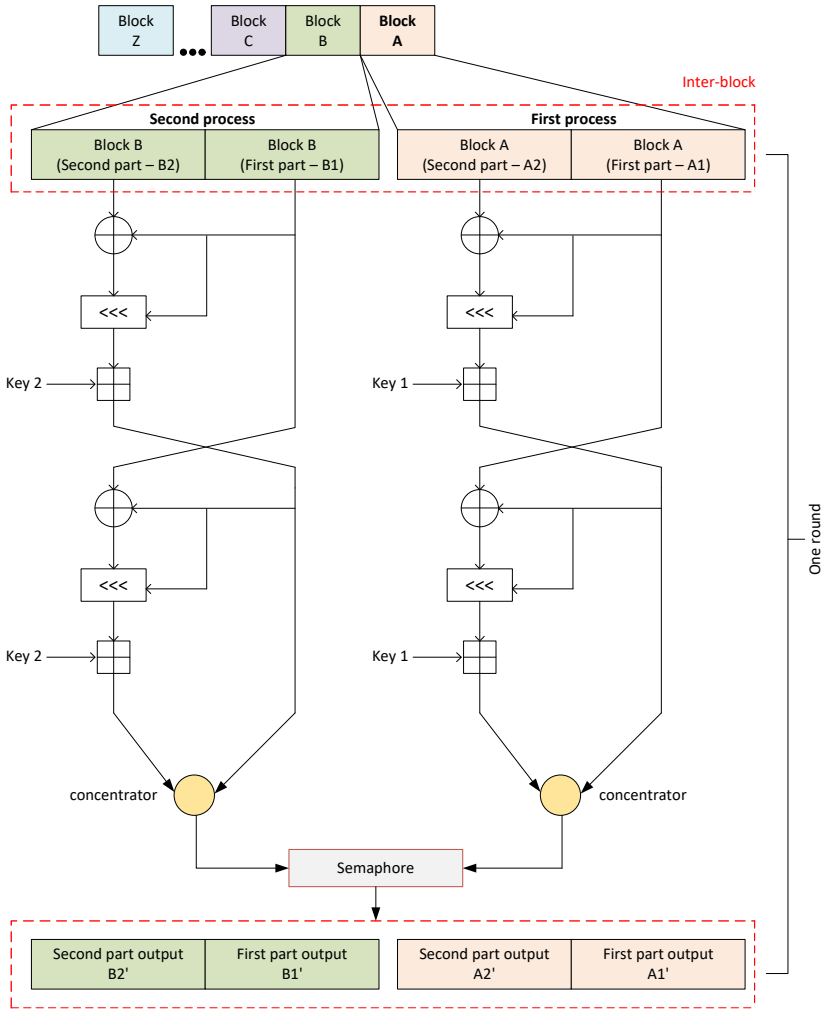


Figure 2: Enhanced RC5 structure

The structure of the enhanced RC5 algorithm needs the use of two keys, one for the first process and another for the second. This is the feature that makes possible to use a key of a larger length. In fact, the extended key can be divided into two parts for using the first part for the first process and the second one for the another.

For instance, because of the maximum key length of the traditional RC5 algorithm is 2040 bits, now working with this scenario we can use a key of 4080 bits for running an inter-block of two RC5 processes each one with a key of 2040 bits. In order to work with a key of more than 4080 bit a third process will need to be included.

The RC5 algorithm has a major advantage because it can use any key length until 2040 bits, at difference to other algorithms as AES which works with just three key lengths. This permits to compare the traditional RC5 algorithm with the enhanced one. For instance, we can measure the duration of time the traditional algorithm is able to encrypt with a key of 128 bits, and in the same way we can measure the time taken by the enhanced algorithm with the same key length applying 64 bits for the first process and other 64 bits for the another.

Although the mathematical operations over the blocks are exactly the same, is possible that the two process do not finish their work at the same the time. For this reason it is necessary to implement a mechanism to handle the simultaneity between the two events.

At each process is carried out logical operations with the data and the respective key. These operations are represented in Algorithm 1 for the block A. The same operations are carried out simultaneously for the block B.

Algorithm 1 Mathematical representation of RC5 encryption

```
A2 := A2 + K[0]
A1 := A1 + K[1]
for i := 1 to r do
    A2 := ((A2 XOR A1) left shift A1) + K[2i]
    A1 := ((A1 XOR A2) left shift A2) + K[2i+1]
end for
```

Although is difficult to model mathematically the relation between the performance of the traditional RC5 algorithm and the enhanced one, an approximate equation for keys larger than 1024 bits derived from the results obtained is presented in Equation 1, where P is the performance of the traditional RC5 algorithm (expressed as the time in microseconds taken for doing the encryption process), k is the number of bits of the key, and P' is the performance of the enhanced one (expressed in microseconds as well). Of course, this is not an exact rule for all the cases.

$$P' \approx \frac{P}{2} + \frac{k}{1024} \times 100.675[us] \tag{1}$$

The decryption process works similar. There will be two processes in order to decrypt the data and the final results in each process will be handled by a semaphore.

3.2 Handling simultaneity

Handling the simultaneity of events is a major task to take into account when working with parallelism. As was shown in Figure 2 the likelihood that the two process of an inter-block finishes their work at different times is handled through a semaphore.

The semaphore will make that the process which finishes its work first waits until the another also finishes. When both processes finish, the semaphore will permit to continue with the following inter-block of data.

In the same way the semaphore will permit to detect when a process fails to complete its work.

The Algorithm 2 explains the functionality of the semaphore. First, the semaphore enters inside an infinite loop in order to wait until the two processes have finished their tasks. When the semaphore detects that both processes have finished then analyzes the results of each

process. If both processes are successful, so the semaphore continue with the next inter-block, if one process has an error, so the process is run again, and if both processes have an error means that both of them are repeated.

Algorithm 2 Semaphore simultaneity handle algorithm

```
1: stopProcess := true
2: while stopProcess is true do
3:     if process1 has finished and process2 has finished  then
4:         stopProcess := false
5:         if process1 is successful and process2 is successful then
6:             continue with next inter-block
7:         else if process1 is not successful and process2 is successful then
8:             repeat process1
9:         else if process1 is successful and process2 is not successful then
10:            repeat process2
11:        else
12:            repeat both processes
13:        end if
14:    end if
15: end while
```

4 Results analysis

As was explained in section 2, several cryptography algorithms can be used in communication networks. Table 1 shows the maximum key length of some symmetric and asymmetric algorithms. As Figure 3 reflects the key of asymmetric algorithms is larger than the symmetric ones. RC5 is a particular algorithm which permits the use of a larger key.

Table 1: Maximum key length used by some algorithms

	Comparison	
	Algorithm	**Maximum key length (bits)**
Asymmetric	RSA	4096
	DSA	3072
Symmetric	AES	256
	RC5	2048
	3DES	168

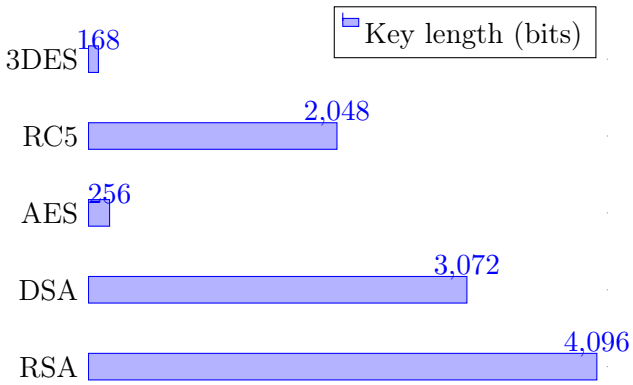


Figure 3: Key length comparison

However, several techniques in order to expand the key length and therefore the security level of symmetric algorithms can be applied. Table 2 summarizes the methods analyzed in section 2 with the number of articles researched that each technique apply. Of course, some articles apply more than just one technique.

Table 2: Cryptographic techniques for performance improvement

Technique	Quantity
Key length expansion	1
Algorithm modifications	4
New hardware architecture	9
Key exchange modifications	4
Parallel Computing	3
Small data	3
Reduced rounds	3
Improved logical operations	8
Quantum Computing	1

Figure 4 explains the fact that the preferred option for improving symmetric key algorithms' performance is through hardware implementations.

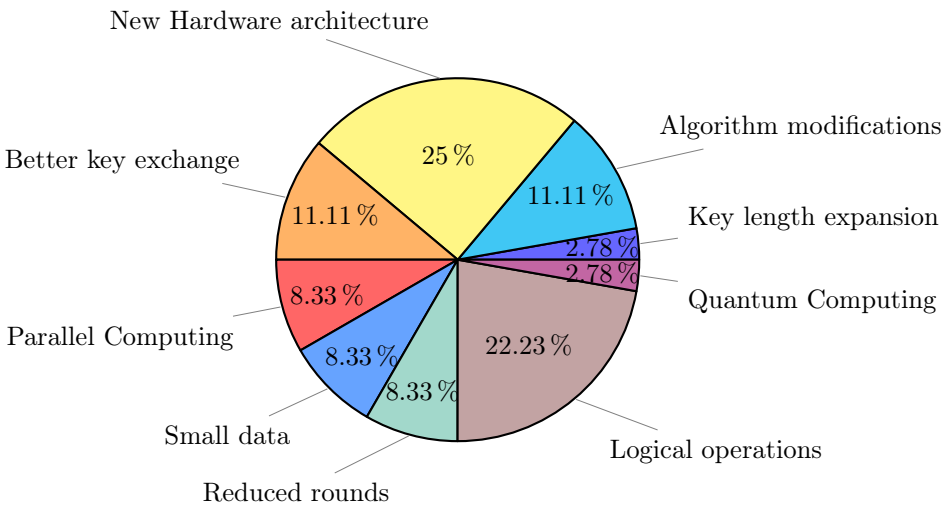


Figure 4: Researched techniques for performance improvement.

Modifying the RC5 algorithm as was explained in section 3 is possible to improve the performance reducing the elapsed time taken by encryption. Table 3 compares the time taken by the traditional RC5 algorithm and the enhanced one considering the key length and the number of rounds used for encryption. As the results in the table explains when the number of rounds increases, the performance of the algorithm decreases due to the additional load the algorithm has to process.

Table 3: Comparison with different number of rounds.

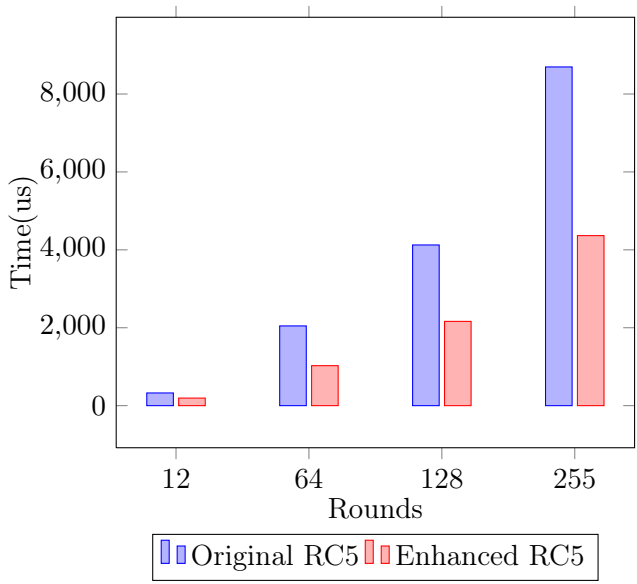
	Comparison		
	Key length(bits)	**RC5(us)**	**Enhanced RC5(us)**
12 rounds	510	64.245	32.241
	1020	150.125	85.471
	2040	325.854	194.587
	4080	not feasible	360.124
64 rounds	510	556.124	256.251
	1020	1025.958	512.632
	2040	2047.147	1026.587
	4080	not feasible	1987.524
128 rounds	510	1030.127	567.254
	1020	2048.487	1087.254
	2040	4125.254	2163.478
	4080	not feasible	4325.125
255 rounds	510	2063.245	1245.487
	1020	4236.147	2145.551
	2040	8695.235	4365.145
	4080	not feasible	8984.258

As Figure 5 shows, for a certain number of rounds the enhanced RC5 algorithm is able to encrypt the same amount of data than the traditional algorithm much faster, using the same key length, due to the use of parallelism.

This better performance is not considerable when the key length is too short, because the expanded key is also short. However, the

performance can be important and considerable when the key length is expanded and this fact can play an important role for certain applications transmitted over communication networks.

Figure 5: Comparison for 2040 bits key length and different number of round.



In Figure 6 is represented the performance of the enhanced RC5 algorithm. As the figure explains, when the key length used is larger the time needed for encryption increases because the algorithm needs to process a major number of bits at each operation, but the security level is enhanced. On the other hand, the figure determines that for a certain key length the performance decreases if the algorithm uses a larger number of rounds due to the algorithm needs more time for the encryption process of the additional rounds.

Using a larger number of rounds the security level provided by the algorithm can be increased because the encrypted data is more vulnerable to lineal cryptanalysis with a minor number of rounds. Therefore, determining the proper key length and an acceptable number of rounds is an important aspect to take into consideration.

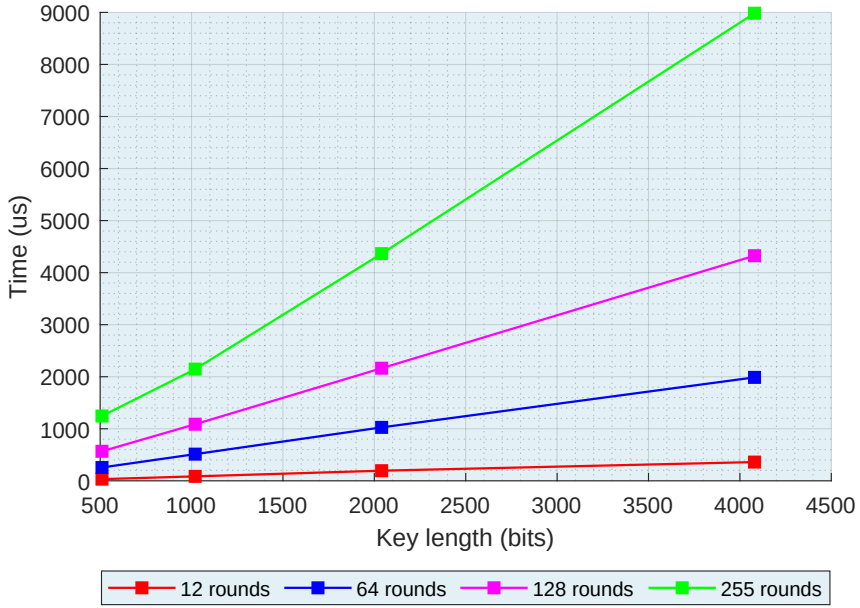


Figure 6: Enhanced RC5 performance for different number of rounds.

Another important factor to take into consideration is the block length used by the algorithm. The RC5 algorithm uses blocks of 32, 64 and 128 bits. As other symmetric key algorithms, the RC5 block length plays an important role during the encryption process and it is important to determine this factor in order to obtain a safe enough algorithm with an acceptable performance.

Table 4 shows the results obtained for the RC5 traditional algorithm and the enhanced one for each block length.

Using a larger block length for encryption permits to use a minor number of inter-blocks to encrypt the data. On the other hand, the RC5 algorithm will have to use a greater number of inter-blocks if the block length is minor, because at each process less data will be encrypted.

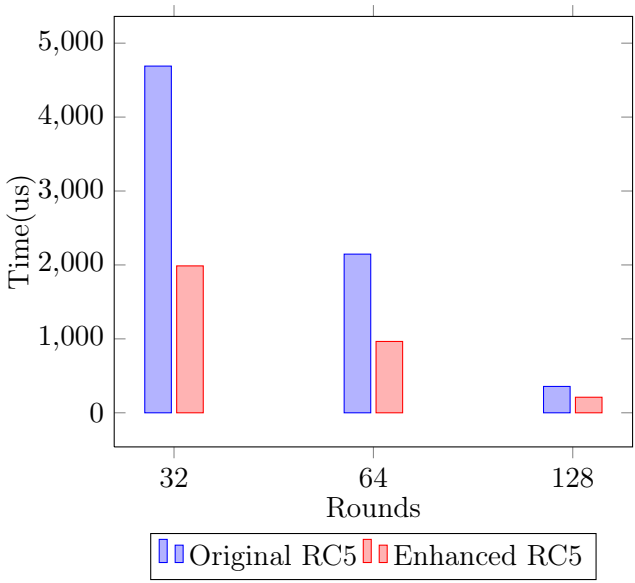


Figure 7: Comparison for 2040 bits key and different block lengths.

Table 4: Comparison with different block lengths.

	Comparison		
	Key length(bits)	**RC5(us)**	**Enhanced RC5(us)**
32 bits	510	1584.127	524.154
	1020	2415.256	998.639
	2040	4689.784	1987.158
	4080	not feasible	4654.145
64 bits	510	675.145	310.547
	1020	1248.632	510.633
	2040	2146.254	965.125
	4080	not feasible	2547.487
128 bits	510	57.524	28.157
	1020	179.256	95.145
	2040	356.587	210.487
	4080	not feasible	487.741

As Figure 7 reflects, for a certain block length the performance of the RC5 enhanced algorithm is better than the traditional one using the same key length. Again, the use of parallelism is an important technique because it permits to make the encryption process in a minor time.

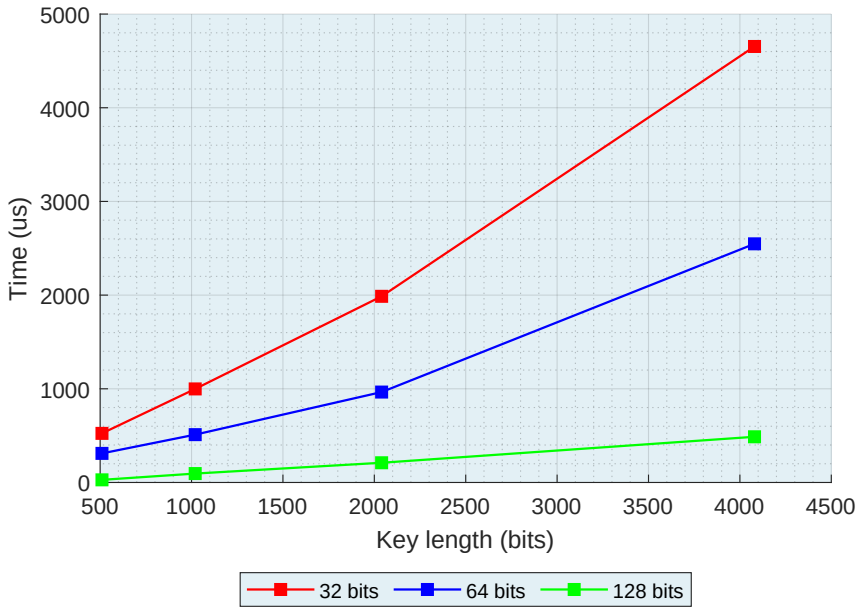


Figure 8: Enhanced RC5 performance for different block lengths.

Figure 8 analyzes the enhanced RC5 algorithm performance when different block lengths are used. The figure shows that for a certain key length the performance of the algorithm is improved when the block length is larger, due to the algorithm is able to encrypt the data in less time avoiding the use of additional inter-blocks. On the other hand, increasing the key length will reduce the performance due to the need to process more bits.

5 Conclusions

The use of Parallel Computing helps to improve the performance of the encryption process by permitting to make two or even more tasks simultaneously. In this case, using Parallel Computing was feasible to modify the RC5 algorithm to work in an inter-block scenario with an enhanced security level achieved by a key expansion technique maintaining an acceptable performance for communication networks.

In comparison with other advanced mechanisms, the use of a key expansion technique in company of the use of parallelism permitted to improve the security level without a degradation in the performance of the RC5 algorithm. This was achieved without the use of advanced High Performance Computing techniques which might not be available in all environments.

Two important factors in the performance of the enhanced RC5 algorithm are the number of rows and the block length used for the encryption process. As the number of rows increases, the encryption process needs also more time, although the security level increases as well. However, the results demonstrates that using blocks of larger lengths the overall process is faster due to the fact that the algorithm does not need more blocks processing in order to encrypt the data.

This demonstrates that is feasible to improve the security level of the RC5 algorithms for actual communication networks using Parallel Computing, and that this mechanism can be used in several applications.

Although this research was focused in the RC5 algorithm enhancement, the concepts and the idea of using inter-blocks units can be applied to other symmetric key algorithms based on cipher blocks.

Of course, other methods in addition to a semaphore might be required to handle events simultaneity depending on the features of each algorithm.

Acknowledgment

A special acknowledgment is expressed to Universidad Tecnológica Equinoccial, which provided the necessary equipment and facilities for the development of this research.

References

[1] S. Ahmad, K. M. R. Alam, H. Rahman, and S. Tamura, "A comparison between symmetric and asymmetric key encryption algorithm based decryption mixnets," *Proceedings of 2015 International Conference on Networking Systems and Security, NSysS 2015*, 2015. [Online]. Available: https://doi.org/10.1109/NSysS.2015.7043532 104, 105

[2] J. S. Coron, "What is cryptography?" *IEEE Security and Privacy*, 2006. [Online]. Available: https://doi.org/10.1109/MSP.2006.29 104

[3] S. Chandra, S. Paira, S. S. Alam, and G. Sanyal, "A comparative survey of symmetric and asymmetric key cryptography," *2014 International Conference on Electronics, Communication and Computational Engineering, ICECCE 2014*, pp. 83–93, 2014. [Online]. Available: https://doi.org/10.1109/ICECCE.2014.7086640 104

[4] J. N. Gaithuru, M. Bakhtiari, M. Salleh, and A. M. Muteb, "A comprehensive literature review of asymmetric key cryptography algorithms for establishment of the existing gap," *2015 9th Malaysian Software Engineering Conference, MySEC 2015*, pp. 236–244, 2016. [Online]. Available: https://doi.org/10.1109/MySEC.2015.7475227 105

[5] A. Mirtalebi and S. M. Babamir, "A cryptography approach on security layer of web service," *Application of Information and Communication Technologies, AICT 2016 - Conference Proceedings*, 2017. [Online]. Available: https://doi.org/10.1109/ICAICT.2016.7991698 105

[6] S. Vyakaranal and S. Kengond, "Performance Analysis of Symmetric Key Cryptographic Algorithms," *2018 International Conference on Communication and Signal Processing (ICCSP)*, pp. 411–415, 2018. 105

[7] B. Mandal, S. Chandra, S. S. Alam, and S. S. Patra, "A comparative and analytical study on symmetric key cryptography," *2014 International Conference on Electronics, Communication and Computational Engineering, ICECCE 2014*, pp. 131–136, 2014. [Online]. Available: https://doi.org/10.1109/ICECCE.2014.7086646 105

[8] M. Peyravian and D. Coppersmith, "Structured symmetric-key block cipher," *Computers and Security*, vol. 18, no. 2, pp. 134–147, 1999. [Online]. Available: https://doi.org/10.1016/S0167-4048(99)90053-6 105

[9] N. Bajaj and A. Thakur, "Enhancement of RC5 for image encryption," *ICIIP 2011 - Proceedings: 2011 International Conference on Image Information Processing*, vol. 2, no. Iciip, pp. 7–11, 2011. [Online]. Available: https://doi.org/10.1109/ICIIP.2011.6108973 106

[10] W. Changda and J. Shiguang, "Multilevel security model for ad hoc networks," *Journal of Systems Engineering and Electronics*, vol. 19, no. 2, pp. 391–397, 2008. [Online]. Available: https://doi.org/10.1016/S1004-4132(08)60098-5 106

[11] T. Nie, Y. Li, and C. Song, "Performance evaluation for CAST and RC5 encryption algorithms," *2010 International Conference on Computing, Control and Industrial Engineering, CCIE 2010*, vol. 1, pp. 106–109, 2010. [Online]. Available: https://doi.org/10.1109/CCIE.2010.34 106

[12] O. Elkeelany and S. Nimmagadda, "Performance Evaluation of Differente Hardware Models of RC5 Algorithm," *ICIIP 2011 - Proceedings: 2011 International Conference on Image Information Processing*, pp. 142–145, 2007. 106

[13] I. A. Kamil and S. O. Ogundoyin, "An improved certificateless aggregate signature scheme without bilinear pairings for vehicular ad hoc networks," *Journal of Information Security and Applications*, vol. 44, pp. 184–200, 2019. [Online]. Available: https://doi.org/10.1016/j.jisa.2018.12.004 107

[14] N. B. Silva, D. F. Pigatto, P. S. Martins, and K. R. Branco, "Case studies of performance evaluation of cryptographic algorithms for an embedded system and a general purpose computer," *Journal of Network and Computer Applications*, vol. 60, pp. 130–143, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.jnca.2015.10.007 107

[15] S. O'Melia and A. J. Elbirt, "Instruction set extensions for enhancing the performance of symmetric-key Cryptography," *Proceedings - Annual Computer Security Applications Conference, ACSAC*, pp. 465–474, 2008. [Online]. Available: https://doi.org/10.1109/ACSAC.2008.10 107, 108

[16] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "Efficient and high-performance parallel hardware architectures for the AES-GCM," *IEEE Transactions on Computers*, vol. 61, no. 8, pp. 1165–1178, 2012. [Online]. Available: https://doi.org/10.1109/TC.2011.125 107

[17] L. Hua, L. Jianzhou, and Y. Jing, "An efficient and reconfigurable architecture for RC5," *Canadian Conference on Electrical and Computer Engineering*, vol. 2005, pp. 1652–1655, 2005. [Online]. Available: https://doi.org/10.1109/CCECE.2005.1557299 107

[18] A. J. Elbirt and C. Paar, "An instruction-level distributed processor for symmetric-key cryptography," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 5, pp. 468–480, 2005. [Online]. Available: https://doi.org/10.1109/TPDS.2005.51 107

[19] Y. Yifeng, G. Yong, W. Heyu, and L. T. School, "A Symmetric Key Exchange Protocol Bsaed on Virtual S-Box," *China Communications*, vol. 11, no. 14, p. 5522, 2014. 107

[20] R. S. Kumar, E. Pradeep, K. Naveen, and R. Gunasekaran, "Enhanced cost effective symmetric key algorithm for small amount of data," *2010 International Conference on Signal Acquisition and Processing, ICSAP 2010*, pp. 354–357, 2010. [Online]. Available: https://doi.org/10.1109/ICSAP.2010.13 108

[21] T. Teerakanok and S. Kamolphiwong, "Accelerating asymmetric-key cryptography using Parallel-key Cryptographic Algorithm (PCA)," *2009 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pp. 812–815, 2009. [Online]. Available: https://doi.org/10.1109/ECTICON.2009.5137170 108

[22] C. Negre and J. M. Robert, "New parallel approaches for scalar multiplication in elliptic curve over fields of small characteristic," *IEEE Transactions on Computers*, vol. 64, no. 10, pp. 2875–2890, 2015. [Online]. Available: https://doi.org/10.1109/TC.2015.2389817 108

[23] P. Bilski and W. Winiecki, "Multi-core implementation of the symmetric cryptography algorithms in the measurement system," *Measurement: Journal of the International Measurement Confederation*, vol. 43, no. 8, pp. 1049–1060, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.measurement.2010.03.002 108

[24] A. Amerimehr and M. H. Dehkordi, "Quantum Symmetric Cryptosystem Based on Algebraic Codes," *IEEE Communications Letters*, vol. 22, no. 9, pp. 1746–1749, 2018. [Online]. Available: https://doi.org/10.1109/LCOMM.2018.2844245 108

[25] J. P. Aumasson, "The impact of quantum computing on cryptography," *Computer Fraud and Security*, vol. 2017, no. 6, pp. 8–11, 2017. [Online]. Available: http://dx.doi.org/10.1016/S1361-3723(17)30051-9 108

[26] H. A. Kholidy, A. A. Azab, and S. H. Deif, "Enhanced "ULTRA GRIDSEC": Enhancing high performance symmetric key cryptography schema using pure peer to peer computational grid middleware (HIMAN)," *2008 3rd International Conference on Pervasive Computing and Applications, ICPCA08*, vol. 1, pp. 26–31, 2008. [Online]. Available: https://doi.org/10.1037/h0094584 108

[27] J. Liang, Q. Wang, Y. Qi, and F. Yu, "An area optimized implementation of cryptographic algorithm RC5," *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 4–7, 2009. 109

[28] E. B. Villanueva, R. P. Medina, and B. D. Gerardo, "An enhanced RC5 (ERC5) algorithm based on simple random number key expansion technique," *ISCAIE 2018 - 2018 IEEE Symposium on Computer Applications and Industrial Electronics*, vol. 5, pp. 134–138, 2018. [Online]. Available: https://doi.org/10.1109/ISCAIE.2018.8405458 109

[29] M. Sharafi, F. Fotouhi-Ghazvini, M. Shirali, and M. Ghassemian, "A low power cryptography solution based on chaos theory in wireless sensor nodes," *IEEE Access*, vol. 7, no. c, pp. 8737–8753, 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2018.2886384 109